

## CLAIMS

1 1. A method of processing a text file in a computer  
2 application, comprising the steps:

3  
4 forming a template from fragments of the text file;  
5 using the template as an overlay for parsing incoming  
6 files, or as a prototype to generate a segment of an  
7 output file;

8  
9 using a macro class to map data from the text file to an  
10 application; and

11  
12 embedding the macro class as a keyword within the  
13 template, wherein when the template reaches the keyword,  
14 the template calls the macro class to further process  
15 the text file.

1 2. A method according to Claim 1, wherein the macro  
2 class reads in a segment of the text file and uses the  
3 segment to initiate application update processing.

1 3. A method according to Claim 1, wherein the macro  
2 class derives data from the application and formats it  
3 into the text file.

1 4. A method according to Claim 1, wherein the macro  
2 class derives a template name from the invoking template  
3 and uses that name to invoke a next template to further  
4 process the text file.

1 5. A method according to Claim 1, further comprising the

step of providing an interface controller to prevent structure clashes by placing text data into appropriate places in a complex object structure as the text file is processed.

6. A system for processing a text file in a computer application, comprising:

means for forming a template from fragments of the text file;

means for using the template as an overlay for parsing incoming files, or as a prototype to generate a segment of an output file;

means for using a macro class to map data from the text file to an application; and

means for embedding the macro class as a keyword within the template, wherein when the template reaches the keyword, the template calls the macro class to further process the text file.

7. A system according to Claim 6, wherein the macro class reads in a segment of the text file and uses the segment to initiate application update processing.

8. A system according to Claim 6, wherein the macro class derives data from the application and formats it into the text file.

9. A system according to Claim 6, further comprising an

001120-6080US1

2 interface controller to prevent structure clashes by  
3 placing text data into appropriate places in a complex  
4 object structure as the text file is processed.

1 10. A program storage device readable by machine,  
2 tangibly embodying a program of instructions executable  
3 by the machine to perform method steps for processing a  
4 text file in a computer application, said method steps  
5 comprising:

6  
7 forming a template from fragments of the text file;  
8 using the template as an overlay for parsing incoming  
9 files, or as a prototype to generate a segment of an  
10 output file;

11  
12 using a macro class to map data from the text file to an  
13 application; and

14  
15 embedding the macro class as a keyword within the  
16 template, wherein when the template reaches the keyword,  
17 the template calls the macro class to further process  
18 the text file.

1 11. A program storage device according to Claim 10,  
2 wherein the macro class reads in a segment of the text  
3 file and uses the segment to initiate application update  
4 processing.

1 12. A program storage device according to Claim 10,  
2 wherein the macro class derives data from the  
3 application and formats it into the text file.

1 13. A program storage device according to Claim 10,  
2 wherein said method steps further comprise the step of  
3 providing an interface controller to prevent structure  
4 clashes by placing text data into appropriate places in  
5 a complex object structure as the text file is  
6 processed.

00615009-074400